



AF 3100

PTO/SB/21 (07-06)

Approved for use through 09/30/2006. OMB 0651-0031

U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

**TRANSMITTAL
FORM**

(to be used for all correspondence after initial filing)

Total Number of Pages in This Submission

Application Number 09/900,079

Filing Date July 6, 2001

First Named Inventor Vandersluis

Art Unit 2161

Examiner Name Nguyen, Cindy

Attorney Docket Number XAW-0102

ENCLOSURES (Check all that apply)

| | | |
|---|---|--|
| <input type="checkbox"/> Fee Transmittal Form | <input type="checkbox"/> Drawing(s) | <input type="checkbox"/> After Allowance Communication to TC |
| <input type="checkbox"/> Fee Attached | <input type="checkbox"/> Licensing-related Papers | <input type="checkbox"/> Appeal Communication to Board of Appeals and Interferences |
| <input type="checkbox"/> Amendment/Reply | <input type="checkbox"/> Petition | <input checked="" type="checkbox"/> Appeal Communication to TC (Appeal Notice, Brief, Reply Brief) |
| <input type="checkbox"/> After Final | <input type="checkbox"/> Petition to Convert to a Provisional Application | <input type="checkbox"/> Proprietary Information |
| <input type="checkbox"/> Affidavits/declaration(s) | <input type="checkbox"/> Power of Attorney, Revocation | <input type="checkbox"/> Status Letter |
| <input type="checkbox"/> Extension of Time Request | <input type="checkbox"/> Change of Correspondence Address | <input checked="" type="checkbox"/> Other Enclosure(s) (please identify below): |
| <input type="checkbox"/> Express Abandonment Request | <input type="checkbox"/> Terminal Disclaimer | Return Post Card |
| <input type="checkbox"/> Information Disclosure Statement | <input type="checkbox"/> Request for Refund | |
| <input type="checkbox"/> Certified Copy of Priority Document(s) | <input type="checkbox"/> CD, Number of CD(s) _____ | |
| <input type="checkbox"/> Reply to Missing Parts/Incomplete Application | <input type="checkbox"/> Landscape Table on CD | |
| <input type="checkbox"/> Reply to Missing Parts under 37 CFR 1.52 or 1.53 | | |

Remarks

Appeal Brief

No fee due according to MPEP 1204.01

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm Name Law Office of Dale B. Halling

Signature

Printed name Dale B. Halling

Date 1/9/2007

Reg. No. 38,170

CERTIFICATE OF TRANSMISSION/MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below:

Signature

Theresa A. Jones

Typed or printed name

Theresa A. Jones

Date 1/9/2007

This collection of information is required by 37 CFR 1.5. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.



PATENT

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

(Appeal Brief-2)

APPLICANT(S): Vandersluis. EXAMINER: Nguyen, Cindy

SERIAL NO.: 09/900,079 ART GROUP: 2161

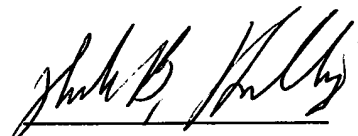
FILED July 6, 2001 Case No.: XAW-0102

ENTITLED: System and Method for Converting Data in a First Hierarchical Data
Scheme into a Second Hierarchical Data Scheme

I hereby certify that this correspondence is being deposited with the United States
Postal Service as first-class mail in an envelope addressed to: Honorable
Commissioner of Patents and Trademarks, P.O. Box 1450, Alexandria, VA 22313-
1450 on:

1/9/06
Date of Deposit

Dale B. Halling
Attorney of Record


Signature

APPEAL BRIEF-2

Honorable Commissioner of
Patents and Trademarks
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This is an appeal from the rejection of claims 1-29 of the Office Action dated October 2, 2006, which was a reopened prosecution in view of the Appeal Brief filed on August 26, 2005. A Supplemental Notice of Appeal was filed on December 22, 2006. The original Appeal was from the rejection of claims 1-29 of the Office Action dated March 23, 2005. This application was filed on July 6, 2001. Appellant submits this Appeal Brief pursuant to 35 U.S.C. §134 and 37 C.F.R. § 1.191 in furtherance of the Notice of Appeal filed in this case on December 22, 2006. The fees required under 37 C.F.R. §1.17(b) and 37 C.F.R. § 41.20(2) and any other necessary fees were paid with respect to the original Appeal Brief filed on August 26, 2005 according to MPEP 1204.01 no additional fees are due, since the fees have not changed since the original appeal brief and notice of appeal were filed.

I. Real Party In Interest

The real party in interest is: XAware Inc., a Colorado corporation, having a place of business at 5555 Tech Center Dr., Suite 200, Colorado Springs, CO 80919. See the Assignment recorded at Reel 012236, Frame 0990.

II. Related Appeals And Interferences

The Original Appeal was from the rejection of claims 1-29 of the Office Action dated March 23, 2005. There are no interferences related to the present appeal.

III. Status Of Claims

Claims 1-29 (see Appendix) are pending in this application. Claims 1-29 are rejected and are involved in this appeal. Note that claim was listed on page 11 of the Office Action dated 10/2/06 as allowable however claim 16 was also rejected under 35 USC 101 in the same Office Action on page 3.

IV. Status Of Amendments

There have been no amendments filed subsequent to the rejection of March 23, 2005.

V. Summary Of The Invention

Claims 1, 13, 18 & 26 are argued separately below. Claim 1 requires a template defining the second hierarchical data scheme. The template is shown in FIG. 1 as element 22, in FIG. 2 as element 44 and FIG. 11 shows an example of a template. According to the specification, page 5, lines 21-26, "The system 20 includes a template 22 that defines a second hierarchical data scheme. In one embodiment the second hierarchical data scheme may be a sample extensible markup language file, an XML target format template, a extensible markup language document type definition or an extensible markup language schema". Claim 1 further requires that the hierarchical data scheme is a scheme that groups data and its context. This is an inherent feature of XML (eXtensible Markup Language) and the other examples of hierarchical data schemes. The next element of claim 1 is a dynamic data generation module. The dynamic data generation module is shown in FIG. 1 as element 24, in FIG. 2 as element 46, 48 & 50. Claim 1 requires that the dynamic data generation module be contained in the template. According to the specification, page 6, lines 2-3 "A dynamic data generation module 24 is contained in the template 22." In addition the specification states, page 6, lines 5-12, "The dynamic data generation module 24 includes a query 28 to the data source 26 and driver for connecting to the data source 26. The results of the query, in one embodiment, place the acquired data in the appropriate part of the template. In another embodiment, the query takes the data from the appropriate part of the template and places it in the data store (26). In this way the data is converted from the first hierarchical data scheme to the second hierarchical data scheme or vice versa." The next element in claim 1 is a data source. A data source is shown in FIG. 1 as element 26 and in FIG. 2 as elements 56, 58. According to the specification page 6, lines 3-5 "A data source 26 is in communication with the dynamic data generation module 24. The data source 26 contains data in the first hierarchical data scheme" as required by claim 1.

Claim 13 requires publishing a dynamic template in a server. According to the specification page 7, lines 24-26, "A wizard 70 (FIG. 2) or set of wizards walk a user through the process of converting the static template 68 (FIG. 2) into a dynamic template 72 (FIG. 2) that is then published to a server 42(FIG. 2). The next step in claim

13 is receiving an instruction from a client at the dynamic template. According to the specification page 7, lines 11-12 "In one embodiment the client transmits a key and an instruction 62 to the server 42 and receives an XML document related to the key 62." The next step in claim 13 requires executing the dynamic template. This is shown as step 76 of FIG. 3. The next step in claim 13 requires when a dynamic data generation module is executed performing a data transfer operation that converts data in the first hierarchical data scheme into the second hierarchical data scheme. According to the specification pages 8-9, lines 25-26 & 1-2 "When a dynamic data generation module is executed at step 78, a data transfer operation is performed that converts data in the first hierarchical data scheme into the second hierarchical data scheme which ends the process at step 80." Claim 13 requires that the hierarchical data scheme is a scheme that groups data and its context which is inherent to XML and the other examples of hierarchical data schemes. According to the specification page 6, lines 12-21 "In one embodiment, the first hierarchical data scheme may be an extensible markup language scheme, a relational database, a non-relational database, an extensible markup language database, or a self describing database. The second hierarchical data scheme, in one embodiment, may be an extensible markup language scheme, a relational database, a non-relational database, an extensible markup language database, or a self describing database. Note that the first and second hierarchical data scheme may be in the same class (e.g., both relational databases) but have a different data scheme."

Claim 18 requires as its first step receiving a static extensible markup language template. This is shown as element 92 of FIG. 4. The next step of claim 18 is determining for each element of the static extensible markup language template if a datum needs to be dynamically generated. This is element 94 of FIG. 4. The next step of claim 18 is when the datum needs to be dynamically generated, receiving a data source having data in the hierarchical data scheme for acquiring the datum. This is step 96 of FIG. 4. The next step of claim 18 is receiving a data map between a data element in the data source and a metatag in the static extensible markup language template. This is step 98 of FIG. 4. The last step of claim 18 is repeating steps (b) through (d) for every element of the static extensible markup language template to form a dynamic data conversion program. This is element 100 of FIG. 5.

Claim 26 requires as its first step receiving a sample extensible markup language file. This is element 112 of FIG. 6. The next step of claim 26 is determining for each element of the sample extensible markup language file if a datum needs to be dynamically processed. This is element 114 of FIG. 6. The next step of claim 26 is when the datum needs to be dynamically processed, receiving an extensible markup language element location for acquiring the datum. This is element 116 of FIG. 6. The next step of claim 26 is receiving a data map between a metatag in the sample extensible markup language file and an element of the hierarchical data scheme. This is element 118 of FIG. 6. The next step of claim 26 is repeating steps (b) through (d) for every element of the sample extensible markup file to form a dynamic data conversion program. This is element 120 of FIG. 7.

VI. Grounds of Rejection to be Reviewed on Appeal

1. Whether claim 1 is unclear under 35 USC 112, second paragraph because of the phrase “a hierarchical data scheme”?
2. Whether claims 1-12 are unclear under 35 USC 112, second paragraph because the claims are silent on the step to realize the data conversion?
3. Whether claims 18 & 26 are unclear under 35 USC 112, second paragraph because of the conditions required for dynamically generated data?
4. Whether claims 1-29 are statutory under 35 USC 101?
5. Whether claims 1-3 & 5-11 are anticipated under 35 USC 102(e) by Draper et al (US 6581062)?
6. Whether claims 4, 12-15, 17-23 & 25-29 are unpatentable under 35 USC 103(a) over Draper et al (US 6581062) in view of Prompt et al (US 6985905)?
7. Whether claim 24 is unpatentable under 35 USC 103(a) over Draper et al (US 6581062) in view of Prompt et al (US 6985905) in view of Povilus (US 5740425)?

VII. Argument

Issue 1 Whether claim 1 is unclear under 35 USC 112, second paragraph because of the phrase “a hierarchical data scheme?”

A claim is not indefinite if “one skilled in the art would understand all the language in the claims when read in light of the specification.” *Rosemount, Inc. v. Beckman Instruments, Inc.*, 727 F.2d 1540, 221 USPQ 1, 7 (Fed. Cir. 1984). The Examiner suggests that claim 1 is indefinite because it is uncertain whether a hierarchical data scheme refers to the first hierarchical data scheme or the second hierarchical data scheme (OA 10/2/06 page 2). The section pointed to by the Examiner is a “wherein” clause which explains that any hierarchical data scheme is a scheme that groups data and its context. Anyone skilled in the art would know that the phrase refers to both the first and second hierarchical data schemes. The rejection is not supported by the law.

Issue 2 Whether claims 1-12 are unclear under 35 USC 112, second paragraph because the claims are silent on the step to realize the data conversion?

A claim is not indefinite if “one skilled in the art would understand all the language in the claims when read in light of the specification.” *Rosemount, Inc. v. Beckman Instruments, Inc.*, 727 F.2d 1540, 221 USPQ 1, 7 (Fed. Cir. 184). The Examiner suggests that “the claims are silent on the ‘required’ step to realize a system for converting data in a first hierarchical data scheme into a second hierarchical data schema.” (OA 10/2/06 page 2) Claims 1-12 are system claims and

therefore do not have steps. How the system converts between a first hierarchical data scheme into a second hierarchical data schema is clearly spelled out in the specification on pages 5-8 with respect to FIGs. 1-2. No one skilled in the art would find claims 1-12 indefinite. The rejection is not supported by the law.

Issue 3 Whether claims 18 & 26 are unclear under 35 USC 112, second paragraph because of the conditions required for dynamically generated data?

A claim is not indefinite if “one skilled in the art would understand all the language in the claims when read in light of the specification.” *Rosemount, Inc. v. Beckman Instruments, Inc.*, 727 F.2d 1540, 221 USPQ 1, 7 (Fed. Cir. 184). The Examiner suggests that in claims 18 & 26 it is unclear in step (b) under what conditions the data is dynamically generated or not. (OA 10/2/06 page 3) The Examiner is confusing the purpose of the specification with the purpose of the claims. The purpose of the claims is not to explain the invention but to define the invention. Whether the data is dynamically generated or not is clearly explained to one skilled in the art on page 10, lines 7-11, steps 94 & 96 of FIG. 4. As would be clear to one skilled in the art dynamically generated data is data that is created or pulled from another source, while static data is data that is contained in the template. No one skilled in the art would find claims 18 & 26 indefinite. The rejection is not supported by the law.

Issue 4 Whether claims 1-29 are statutory under 35 USC 101?

“Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof,

may obtain a patent therefore” 35 USC 101. The present invention is directed to a problem that industry spends billions of dollars on each year trying to solve, namely allowing different computer systems to talk to each other. More specifically the invention is directed to converting the data in a first format into data in a second format in a computer system or systems. So clearly, the invention is useful. The invention is directed to a process and machine. The invention converts data in a first hierarchical data scheme into a second hierarchical data scheme. This is clearly a process by any definition of the word process. The invention is also a machine. It is an electronic circuit that just happens to be a general purpose computer or computers that are configured into a specific electronic circuit using software to solve a specific problem.

The Examiner suggests that there is not a tangible result that has real world value. The investors in the present assignee and the customers of the assignee of the present invention would be amazed that they had paid good money to obtain a “non-real world” value. The present invention clearly has a tangible result, whether looked at from an economic point of view or from a physical point of view. From an economic point of view the present invention is directed to a portion of a problem that commercial companies have been trying to solve since the first computers were deployed. From a physical point of view a general purpose computer running software is a specific machine or electronic circuit. Electronic circuits are clearly patentable and the present invention is an electronic circuit. The computer running the specific software causes switches to open and close and electronic charges to be stored. This is clearly a physical process and therefore a machine.

The rejection under 35 USC 101 has no support in law or logic and must be withdrawn.

Issue 5 Whether claims 1-3 & 5-11 are anticipated under 35 USC 102(e) by Draper et al (US 6581062)?

“A rejection for anticipation under section 102 requires that each and every limitation of the claimed invention be disclosed in a single prior art reference.” *In re Paulsen*, 30 F.3d 1475, 31 USPQ2d 1671, 1673 (Fed. Cir. 1994).

Claim 1 requires a dynamic data generation module contained in the template. The Examiner suggests (Office Action dated 10/2/06, page 4) that element 52 is the template and that element 50 of Draper is the dynamic data generation module. However, FIG. 1 of the specification clearly shows that the analogy fails because element 50 (the dynamic data generation module – in the Examiner’s analogy) is not contained in element 52 (the template– in the Examiner’s analogy) as required by claim 1. In addition, the template of the present application is the device for converting between the first and second hierarchical data schemes. So the analogy fails because the mapper 50 of Drapper is the device for converting between different formats. Drapper states that FIG. 5 is a flow chart of the mapper and clearly the mapper does not define the second hierarchical data scheme. The Examiner has failed to find a single reference that shows each and every limitation of the claimed invention. Claim 1 is clearly allowable.

Claims 2-3 & 5-11 are allowable as being dependent upon an allowable base claim.

Issue 6 Whether claims 4, 12-15, 17-23 & 25-29 are unpatentable under 35 USC 103(a) over Draper et al (US 6581062) in view of Prompt et al (US 6985905)?

The question of obviousness requires that we determine if the references, taken as a whole, would suggest the invention to one of ordinary skill in the art. *Medtronic, Inc. v. Cardiac Pacemakers, Inc.*, 721 F.2d 1563, 220 USPQ 97 (Fed. Cir. 1983).

Claim 13 requires executing a dynamic template. A dynamic template, according to the specification, is a template of one of the hierarchical data schemes that includes a dynamic data generation module. Neither Draper or Prompt show a dynamic template. The Examiner suggests (Office Action dated 10/2/06, page 4) that element 52 is the template and that element 50 of Draper is the dynamic data generation module. However, FIG. 1 of the specification clearly shows that the analogy fails because element 50 (the dynamic data generation module – in the Examiner's analogy) is not contained in element 52 (the template– in the Examiner's analogy) as required by claim 1. In addition, the template of the present application is the device for converting between the first and second hierarchical data schemes. So the analogy fails because the mapper 50 of Drapper is the device for converting between different formats. Drapper states that FIG. 5 is a flow chart of the mapper and clearly the mapper does not define the second hierarchical data scheme. Prompt is cited because it shows a developer module or wizard. However, the combination of Drapper and Prompt taken as a whole, would not suggest the invention to one of ordinary skill in the art a dynamic template. Claim 13 is allowable.

Claims 4, 12, 14-15 and 17 are allowable as being dependent upon an allowable base claim.

Claim 18 requires determining for each element of the static extensible markup language template if a datum needs to be dynamically generated. The Examiner suggests (Office Action dated 10/2/06, page 4) that element 52 is the template and that element 50 of Draper is the dynamic data generation module. However, FIG. 1 of the specification clearly shows that the analogy fails because element 50 (the dynamic data generation module – in the Examiner's analogy) is not contained in element 52 (the template– in the Examiner's analogy) as required by claim 1. In addition, the template of the present application is the device for converting between the first and second hierarchical data schemes. So the analogy fails because the mapper 50 of Drapper is the device for converting between different formats. Drapper states that FIG. 5 is a flow chart of the mapper and clearly the mapper does not define the second hierarchical data scheme. Prompt is cited because it shows a developer module or wizard. However, the combination of Drapper and Prompt taken as a whole, would not suggest the invention to one of ordinary skill in the art a determining if a datum needs to be dynamically generated or a static template. Claim 18 is allowable.

Claims 19-23 & 25 are allowable as being dependent upon an allowable base claim.

Claim 26 determining for each element of the static extensible markup language template if a datum needs to be dynamically processed. The Examiner

suggests (Office Action dated 10/2/06, page 4) that element 52 is the template and that element 50 of Draper is the dynamic data generation module. However, FIG. 1 of the specification clearly shows that the analogy fails because element 50 (the dynamic data generation module – in the Examiner's analogy) is not contained in element 52 (the template– in the Examiner's analogy) as required by claim 1. In addition, the template of the present application is the device for converting between the first and second hierarchical data schemes. So the analogy fails because the mapper 50 of Drapper is the device for converting between different formats. Drapper states that FIG. 5 is a flow chart of the mapper and clearly the mapper does not define the second hierarchical data scheme. Prompt is cited because it shows a developer module or wizard. However, the combination of Drapper and Prompt taken as a whole, would not suggest the invention to one of ordinary skill in the art a determining if a datum needs to be dynamically processed. Claim 26 is allowable.

Claims 27-29 are allowable as being dependent upon an allowable base claim.

Issue 7 Whether claim 24 is unpatentable under 35 USC 103(a) over Draper et al (US 6581062) in view of Prompt et al (US 6985905) in view of Povilus (US 5740425)?

Claim 24 is allowable as being dependent upon an allowable base claim.

Note that claim 16 is allowable according to the Office Action dated 10/2/06 page 11.

IX. Appendix Of The Appealed Claims

1. A system for converting data in a first hierarchical data scheme into a second hierarchical data scheme, comprising:

a template defining the second hierarchical data scheme, wherein a hierarchical data scheme is a scheme that groups data and its context;
a dynamic data generation module contained in the template; and
a data source, in communication with the dynamic data generation module, containing data in the first hierarchical data scheme.

2. The system of claim 1, wherein the template and the dynamic data generation module are contained in a server.

3. The system of claim 2, further including a driver connected between the dynamic data generation module and the data source.

4. The system of claim 3, further including a developer module contained in the server for creating the dynamic data generation module.

5. The system of claim 1, wherein the template is a static extensible markup language document.

6. The system of claim 1, wherein the template is an extensible markup language document type definition.

7. The system of claim 1, wherein the template is an extensible markup language schema.

8. The system of claim 1, wherein the first hierarchical data scheme is selected from the group of: extensible markup language schemes, relational databases, non-relational databases, extensible markup language databases and self describing databases.

9. The system of claim 1, wherein the second hierarchical data scheme is selected from the group of: extensible markup language schemes, relational databases, non-relational databases, extensible markup language databases and self describing databases.

10. The system of claim 1, wherein the dynamic data generation module includes a query directed to the data source.

11. The system of claim 1, wherein the dynamic data generation module includes a data mapping between the first hierarchical data scheme and the second hierarchical data scheme.

12. The system of claim 4, wherein the developer module contains a wizard that walks a user through a process of creating the dynamic data generation module.

13. A method of converting data in a first hierarchical data scheme into a second hierarchical data scheme, comprising the steps of:

- a) publishing a dynamic template in a server;
- b) receiving an instruction from a client at the dynamic template;
- c) executing the dynamic template; and
- d) when a dynamic data generation module is executed, performing a data transfer operation that converts data in the first hierarchical data scheme into the second hierarchical data scheme, wherein a hierarchical data scheme is a scheme that groups data and its context.

14. The method of claim 13, wherein step (a) further includes the steps of:
- a1) receiving a template;
 - a2) determining for each element of the template if a dynamically generated data is required;
 - a3) when the dynamically generated data is required, receiving a data source for obtaining the dynamically generated data.
15. The method of claim 14, further including the steps of:
- a4) receiving a data mapping between the first hierarchical data scheme and the second hierarchical data scheme.
16. The method of claim 15 wherein step (a4) further includes the steps of:
- i) when the first hierarchical data scheme is a non-extensible markup language and the second hierarchical data scheme is a second non-extensible markup language, creating a first data mapping between the first hierarchical data scheme and an intermediate extensible markup scheme;
 - ii) creating a second data mapping between the intermediate extensible markup scheme and the second hierarchical data scheme.
17. The method of claim 15, further including the step of"
- a5) receiving a key associated with the data mapping.

18. A method of converting data in a hierarchical data scheme into an extensible markup language scheme, comprising the steps of:

- a) receiving a static extensible markup language template;
- b) determining for each element of the static extensible markup language template if a datum needs to be dynamically generated;
- c) when the datum needs to be dynamically generated, receiving a data source having data in the hierarchical data scheme for acquiring the datum;
- d) receiving a data map between a data element in the data source and a metatag in the static extensible markup language template; and
- e) repeating steps (b) through (d) for every element of the static extensible markup language template to form a dynamic data conversion program.

19. The method of claim 18, wherein step (a) further includes the step of receiving a template selected from the group including: an extensible markup language document type definition and an extensible markup language schema.

20. The method of claim 18, wherein step (a) further includes the step of:

- a1) defining an input parameter.

21. The method of claim 18, wherein step (c) further includes the step of:

- c1) receiving a driver.

22. The method of claim 18, wherein step (c) further includes the step of:

- c1) generating a query to the data source.

23. The method of claim 18, wherein step (d) further includes the step of:

d1) receiving a screen having a list of elements from the data source and a list of metatags from the static extensible markup language template.

24. The method of claim 18, wherein step (c) further includes the step of:

c1) displaying an incomplete version of a dynamic extensible markup language template wherein a static element is shown in a first color and a dynamic element is shown in a second color.

25. The method of claim 18, further including the steps of:

e) publishing the dynamic data conversion program to a server;

f) when a query is received at the server for the dynamic data conversion program, executing the dynamic data conversion program to form an extensible markup language document.

26. A method of converting data in an extensible markup language scheme into a hierarchical data scheme, comprising the steps of:

a) receiving a sample extensible markup language file;

b) determining for each element of the sample extensible markup language file if a datum needs to be dynamically processed;

c) when the datum needs to be dynamically processed, receiving an extensible markup language element location for acquiring the datum;

d) receiving a data map between a metatag in the sample extensible markup language file and an element of the hierarchical data scheme; and

e) repeating steps (b) through (d) for every element of the sample extensible markup file to form a dynamic data conversion program.

27. The method of claim 26, wherein step (a) further includes the step of:

a1) defining a key.

28. The method of claim 26, wherein step (d) further includes the steps of:

d1) receiving a query type;

d2) generating a query.

29. The method of claim 28, wherein step (d1) further includes receiving an insert query type.

IX. Evidence Appendix

None

X. Related Proceedings Appendix

None

Respectfully submitted,
(Vandersluis)

By: 

Attorney for the Applicant
Dale B. Halling
Registration No. 38,170
Phone: (719) 447-1990
Fax: (719) 447-9815